Migration

Migration is the process of transferring applications or parts of applications (objects), either online or in batch, from one location (status) to another. Three PAC entities are used for the migration process:

- Migration event
- Object list
- Audit report

As an application moves through its test plan or life-cycle, PAC uses defined procedures and controls to migrate its objects from one linked status to another. These procedures and controls are defined in the migration path and migration event.

This chapter covers the following topics:

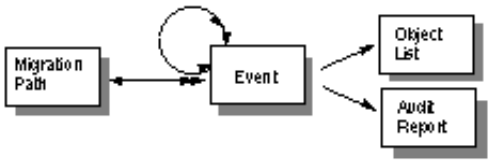
- General Procedure
- Defining Migration Events
- Creating the Object List
- Validating the Object List
- Authorizing a Migration Event
- Submitting a Migration Event
- Monitoring a Migration Event
- Audit Report

General Procedure

The procedure for migrations must be followed in sequence:

1. Define the migration event to migrate the application along an authorized migration path from one status to another.
2. Create the list of objects to be migrated.
3. Validate the object list. PAC checks the list of objects to be migrated.
4. Authorize the event; that is, grant permission to execute the migration.
5. Submit the event; this activates the job that performs the migration.
6. Monitor the processing of the migration event. The Audit Report and the Job Information screens show the progress of the migration.

Defining Migration Events



A migration event is used to execute a migration. An event can migrate any combination of Natural and foreign objects using the same event or separate events; Predict objects must use separate events. A migration event that loads Predict objects into PAC must be run before the event that migrates the application objects that use the Predict objects as subordinates. See Migrating Predict Objects.

An event can only be used for a single migration. Because all actions performed during a specific migration are logged in an audit report, a unique event definition is required to identify this historical information for retrieval at a later date.

It is possible to copy a used event definition to a new event; however, the audit report for the original (copied) event is not copied to the new event. It is also possible to refresh an existing event and reuse the same event definitions. If an event is refreshed, however, all audit information for the existing event is purged.

The migration event definition identifies the information needed to process and control a migration. It specifies

- the migration path (see Defining Migration Paths);
- the list of objects to be migrated (see Creating the Object List);
- the relationship of the event to be migrated with other events;
- the schedule for immediate or future processing.

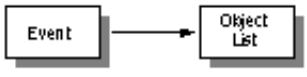
In addition, it may be possible to override some of the migration path specifications.

Scheduling the Migration

You may want to run the migration immediately, or schedule it for a later date/time after another task has been completed.

When migrating to a production status, scheduling allows you to prepare, submit, and process the event at your convenience. The objects for the migration are not implemented into the production environment until the migration is activated by the PAA administrator at the scheduled time. This minimizes outage time because part of the work will already have been completed.

Creating the Object List



The object list comprises objects selected for processing during a migration. The list may

- include one, all, or any number of the application’s objects;
- contain the name of specific objects or identify ranges of objects;
- be manually created or automatically generated;

Types of Objects

PAC and PAA support both saved (source) and cataloged (compiled) objects. A saved object is a Natural, foreign or Predict object created and saved by an application programmer using a source code editor. A cataloged object is a saved object that has been compiled and contains executable code, either Natural or foreign.

Not all objects will exist in compiled form; for example, copycode, text, macros, and recordings exist only in source form. Programs also may exist in source form if the PAC dynamic source option is in use.

For foreign objects, the existence of source and compiled code is predefined by the PAC administrator for each class of object.

Object lists can include

- Predict objects only (see Migrating Predict Objects);
- Natural objects only;
- foreign objects only;
- both Natural and foreign objects.

Foreign Objects for the Object List

PAC provides a mechanism to migrate, version, and audit foreign objects. Predict Application Audit (PAA) tracks foreign objects that are loaded into production.

Before foreign objects can be migrated in PAC, the PAC administrator must define the foreign object datasets to PAC. Once the datasets are defined to PAC, specific datasets can be defined for an application. Thus, the objects that can be migrated are those that belong to the datasets defined for the specific application.

The supported datasets are two formats: source and load (compiled). The default dataset type is source if a dataset type is not specified. foreign objects are not compiled or recompiled when they enter the PAC environment.

The foreign objects that are supported by PAC are user-defined and can include, for example:

```
Assembler copy code
Assembler macro
C
COBOL copy code
COBOL program
JCL
```

Foreign objects are identified on the object list by the prefix 3 on the object type (for example, 3COB is a Cobol object type); this prefix is automatically assigned by PAC.

Although the Expand function is not valid for foreign objects, a Natural program can be expanded to include foreign objects that are used by that Natural program.

Refer to later sections of this documentation for detailed information about migrating foreign objects.

Objects Containing Dynamic Source Variables

Natural programs that contain dynamic source variables (that is, fields prefixed with a plus sign (+) or an ampersand (&)) are referred to as dynamic source. In most cases, programs that contain fields with an & variable must be run, not executed; that is, the program must exist in source form.

PAC requires that you add an extension S to the object type for objects that are to be designated as dynamic source when you create the object list for migrations from development or maintenance. This extension indicates to PAC that the specified object version is to be flagged and migrated as dynamic source for subsequent migrations to test or production statuses; the extension need not be specified for these migrations. Each time the object is migrated from development or maintenance, however, it must be newly flagged to ensure that the new version of the object will be migrated as dynamic source.

Since these objects may require subordinates when they are run, the subordinates (data areas, maps, copycode) must be included in the object list as well.

Note the following compile considerations for subordinates and the states that are required when objects that use these are migrated as dynamic source:

- Programs are not compiled; only the source is migrated;
- Maps are compiled and both source and cataloged forms are migrated;
- Copycode source is migrated;
- Data areas are compiled; only the cataloged form is migrated (Applymod 8 is required).

The following is an example list of objects, including subordinates, that would be supported in the source form for use with dynamic source variables:

```
MENU,PS
MAP1,MS
COPY1,CS
A*,PS
A*,MS
```

where S (the second byte of the second parameter indicating the object type) indicates that this object is to be flagged for migration as dynamic source.

Applymods

The following applymods are provided for use with dynamic source:

- Applymod 2
  Used for syntax table delivery.
- Applymod 8
  Allows all views/data areas to be unloaded during migration because data areas and views may be needed to run the dynamic source programs at execution time.

  Note:
  If programs are not compiled, PAC is unable to determine the subordinate usage of this object; therefore, the expand will not work.
- Applymod 19
  Forces source and cataloged objects to be migrated for all objects in the object list; however, the objects will not be treated as dynamic source and will be compiled normally.
- Applymod 31
  Used for migrating objects.

Options for Building an Object List

You can either generate or "manually" create a list of objects to be migrated.

- If you want to migrate only a few objects, you may enter the individual objects in the Object List Editor.
- If you want to migrate large numbers of objects, you may use the generation facilities. This method avoids the errors that could occur when you manually enter objects.

PAC provides different methods for generating object lists to accommodate different types of migrations or different site requirements. Applymods, user exits, and APIs are available to further customize object lists.

See Creating and Validating a Migration List for detailed information about building the object list for specific types of object migrations.

Manually Creating the Object List

When you create an object list "manually" in the Object List Editor, you may enter objects in different forms to accommodate the type of migration. For example, you may enter

- the specific object name (or range notation);
- the object type (or range notation);

  Note:
  Currently, range notation is not available for selecting foreign objects.

- an optional reference such as a status name or version number. The status name is effective because it uses the version of the object present in the particular execution environment represented by the status name.

Generating the Object List

Note:
You can generate a list of foreign objects only where the origin (From) status is the Control status.

Automation of object grouping guarantees that the list of objects migrated from status to status is always a consistent, related set of objects.

You may automatically generate an object list either online or in batch:

- When you are adding a migration event online, you may generate the object list online by entering a valid value in the Generate List field of the Add Event screen.
- When you are adding a migration event in batch, you may specify a valid value for the GENTYPE option of the ADD EVENT command.

Either online or in batch, you may create the object list for an existing event by issuing the GENLIST command when you are modifying the event. GENLIST may also be used to replace or regenerate an object list for an existing migration event.

Selecting and Expanding Objects for the Object List

You may use the select options (SEL command) to add objects to the object list, either alone or in conjunction with the expand options (EXP command) to include subordinate objects that use or are used by the selected objects.

For foreign objects, only the "used by" expand option is valid. Only Natural objects can be expanded to include subordinate foreign objects that are used by that Natural object.

The SEL Command

The SEL command can be used to select

- Natural objects;
- foreign objects;
- Natural user error messages;
- rules and views to be used during the migration compile phase (of Natural objects).

If selection criteria is provided, the objects that meet the criteria are presented in a list. You may then select those objects to be included in the object list.

The EXP Command

The Expand Status option may be used to include additional objects that may be required during the compile phase.

The object list can be expanded using the Expand option, which can be invoked either manually or automatically.

The Expand option can be used to

- include subordinate objects used by the selected object at compile time;
- include referenced objects used by the selected objects at run time;
- include both subordinate and referenced objects that are used by the selected object;
- include objects that use the selected object:

Objects Used By. When the expand option (EXP command) is requested, PAC takes each entry in the object list in turn and retrieves its subcomponents (for example, views, copycode, data areas, and rules) and adds them to the object list. External referenced foreign objects that are called from a Natural program are also added to the object list.

Objects That Use. (Not valid for foreign objects) For example, if a particular copy code has been changed, you may need to recompile all objects that use the copy code to incorporate and synchronize these changes.

You may optionally reissue the EXP command so that these subcomponents are expanded in turn, the list sorted, and all duplicate objects eliminated. This recursive EXPAND can be used at execution time if Applymod 13 is active in your environment.

Note:
The Expand option may also be used as an option in the migration path definition. This ensures that the subcomponents to be included in the object list are determined based on the overall PAC environment at the time the migration event is processed.

When migrating objects from development or maintenance status types, the Expand function is valid only if the cross-reference data exists for the object on the Predict file specified for the application status link. The EXP (Expand) command and special options of the SEL (Select) command use cross-reference (Xref) data from Predict for migrations from development, maintenance, and incorporation status types; for all other migrations, cross-reference data from PAC is used.

Customizing the Object List

When you have created the object list, you can use system applymods, user exits, and APIs to further customize the objects that are to be migrated.

Applymods

The PAC administrator can choose from a number of applymods to override object list defaults:

- You can suppress the auto expansion of the global data area (GDA) during migration to reduce the number of objects involved in the migration.
- You can instruct PAC to compare the versions of objects added during processing of the Expand option and update the object list with the highest version found. In cases when an object is referenced more than once, this clearly identifies the version to be migrated.
- You can expand the object list recursively. See the section The EXP Command earlier in this section.

User Exits

The PAC administrator can set up user exits to add options for building object lists:

- User Exit 10 can be used to limit the list of objects verified when PAC validates the objects to be processed for a migration event. It can also be used to perform some additional processing on the objects.
- User Exit 23 allows you to selectively suppress expansion or modify Expand option parameters for particular objects.
- User Exit 26 allows you to modify the application general default criteria for selecting objects to be archived.

Application Program Interface (APIs)

The PAC administrator can set up the Object List API in your environment so that you can directly add, purge, or display objects from the object list for a particular migration event.

Validating the Object List

Once created, an object list must be validated before a migration event can be authorized and processed.

You may choose to validate the list immediately, or later, before the event itself becomes authorized. PAC always validates the object list again at execution time to accommodate changes that may have occurred since the list was previously validated.

Depending on the type of migration, PAC performs different types of validation. The following types of questions require answers in any case:

- Does the object exist at the origin location?
- If a range of objects is specified, does at least one object in the range exist at the origin location?
- If an object type (for example, program) is specified to limit the objects in a range, does at least one object of that type in that range exist at the origin location?

If status is used as a reference, PAC verifies this at execution time, and the corresponding version number is substituted before processing continues. The status reference validation ensures that the correct version is used for event processing since the version number may have changed by that time.

How PAC Validates an Object List

PAC uses the following process to validate an object list:

- For objects being copied in from development or maintenance, PAC checks the specified Natural library (or dataset for foreign objects) to ensure that each named object exists and is the correct type.
- For objects being copied from Control, PAC checks against other objects in Control.
- For objects being copied to development or maintenance without replacement, PAC checks the destination library to ensure that the object does not already exist there.
- If a type code has not been provided for a Natural object, PAC updates the object list with the object type found on the system file (origin status).
- If the type code is not consistent with the object type, it is flagged and a warning message is returned.

Revalidation During Processing

Once the object list is validated, the event may be authorized and submitted for processing. During the initial step of the migration process, PAC again validates the object list to determine if all entries are valid.

Note:
Between authorizing the event and processing the event, some time may elapse during which the status of an application and/or any of its objects may change.

Thereafter, if PAC finds an error during the processing of the event, it prints an audit message to the audit report, ignores the object, and then proceeds with processing.

Authorizing a Migration Event

An event must be authorized before it is submitted.

Event authorization is a restricted function that must be performed before an event is submitted. The list of authorizers is specified uniquely for each migration path defined to PAC and is maintained through the Modify Migration Path function.

Once authorized, a migration event is placed in a "frozen" state and cannot be modified. This helps to ensure that only the correct actions and activities take place during the migration.

A migration event, however, can be unauthorized, modified, and then authorized again. A migration event that is copied is automatically unauthorized.

Initiating the Authorization Process

If you have been designated as a migration event authorizer, you can authorize the event online, in batch, or by using the event authorization API provided by PAC if it has been set up in your environment.

- To authorize an event online, enter the AUTHORIZE command on the command line of the Migration Event Authorization screen or use the PF5 key.
- To authorize an event in batch, issue the AUTHORIZE EVENT batch command.

Only an event that is in the Pending or Validated state can be authorized. The authorization request will take a Pending event through the Validated state automatically.

If the event authorization API has been set up in your environment, ask your PAC administrator for information about invoking it.

Customizing the Migration Event Process

When authorizing a migration event, you can set system applymods, which override PAC default processing procedures.

You may override certain PAC defaults during the migration by invoking specific PAC system applymods, user exits, and APIs. See the PAC Reference documentation for detailed information.

Applymods

Applymods can be used to override certain PAC defaults during migrations. For example:

- To conserve space, you may suppress the creation of the audit report during migration (applymod 4). For example, you may only need the report listing from the batch job for verification. The disadvantage is that the suppressed report is not available for an audit review in the PAC system at a later time.
- When copying an event or performing a refresh, you can retain the job name specified in the source event rather than use the job name specified in the migration path. The information is rippled forward and the copied event is run using the same job (applymod 20).

User Exits

User exits can be used to

- generate and verify migration event names;
- add controls at the initialization of the event before migration processing;
- implement audits and security;
- prevent the termination of the migration event after the compile processing step, even if one or more objects do not compile successfully;
- permit the migration unload/load jobs to end normally in spite of certain error conditions being detected.

Application Program Interface (APIs)

The PAC Express APIs provide a useful base for creating a custom system to process migration events quickly and easily. The following APIs are available for migration events:

- The Migration Event API allows you to add, modify, submit, purge or display information about a particular migration event.
- The Object List API allows you to add, purge, or display (retrieve) object list objects for a particular migration event.
- The Text Objects API allows you to display the audit report for a particular migration event.

Submitting a Migration Event

Once a migration event has been defined, details of the event such as the object list must be validated (or optionally overridden). Before a migration event can be submitted, it must have been authorized by an authorizer designated on the migration path.

Based on the options specified at authorization time, the event is submitted for processing either in batch using the remote job entry (RJE) facility, or online in your current Natural session. If an event is submitted online and a job (JCL) has been specified for the migration path, the job is ignored.

You can submit migration events online by entering the SUBMIT direct command on the command line of the Migration Event Submission screen or by using the PF key assigned to confirm submission (PF10).

If the event submission API has been set up in your environment, ask your PAC administrator for information about invoking it. For more information, refer to the section Customizing Migration Processing later in this section.

Monitoring a Migration Event

Once a migration event has been started, the application is locked and no other event for this application can be processed until the first migration event has been completed and unlocked. This restriction ensures that multiple events do not run concurrently, which could cause inconsistent results.

If migration event processing ends abnormally, you have the option to either rerun the event immediately, or to back it out, modify it, and then restart it.

If you decide to back out the incomplete event, any changes made are reversed and the migration event itself is updated unless the event has reached a point of no return (Step 6).

Once processing of a migration event has been completed or backed out, the event may not be processed again or modified unless the REFRESH command is used.

The REFRESH command makes the event reusable by removing all audit information and resetting totals set during its processing.

Locking

The PAC locking mechanism has been enhanced for better data consistency within the migration event process. during migration events into PAC, either from a development or maintenance status it is possible only to execute one migration event at a time, regardless of whether the applications involved are using unique (PRD) applications.

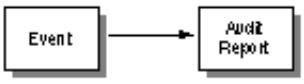
During migrations within or out of PAC it is possible to execute migration events in parallel regardless of whether they use a common (PRD) application or a unique (PRD) application.

This has been changed due to the new way that Predict objects are handled. All Predict objects are now handled within the PCF file and as there is only one PCF file per system this mechanism of locking is essential.

Customized Migration Processing

Migration event processing can be customized using PAC system applymods, user exits, and APIs.

Audit Report



An audit report is a log of the actions performed during a specific migration. It includes the audit, warning, and caution messages needed to verify the results of a migration. The audit report is typically used for problem determination if a migration should fail.

For both online and batch migration events, PAC tracks all migration activities and stores them in an audit report, which is available for review during or after the migration process. In addition, totals of objects processed may be viewed in the job totals.

During processing, PAC may issue a number of warnings and cautions. You therefore need to verify in the audit report that the correct actions have been performed.

